## SIC machine: (Simplified Instructional Computer)

There are two versions of the SIC machine, "simple" SIC and SIC/XE (extended environment).

For SIC, memory is organized as a sequence of 8-bit bytes, and any 3 consecutive bytes forms a word. This means that SIC is designed as a 24-bit machine. A word is addressed by is lowest numbered byte (i.e., addressing starts at byte 0).

Simple SIC:

**Memory**: $2^{15}$ (32 K) bytes

**Registers**:

| mnemonic | number | |
|---|---|---|
| A | 0 | accumulator |
| X | 1 | index register |
| L | 2 | link register |
| PC | 8 | program counter |
| SW | 9 | status word |

**Data formats**:

Numeric - 24 bit 2's complement
Character - 8 bit ASCII

**Instruction format**:

one address instruction architecture, 24 bits as follows

| 8 | 1 | 15 |
|---|---|---|
| op code | x | address |

*index bit*

x = 0 ⇔ direct addressing mode
x = 1 ⇔ indexed/direct addressing mode

**I/O**:

Each device has an 8-bit address; data is transferred in single byte quantities to or from the rightmost byte of register A.

Remark: the SIC simulator on Osprey has as installed devices the 8-bit addresses

$$\underset{\text{boot}}{00},\ \underbrace{04,\ 05,\ 06}_{\text{output}},\ \underbrace{F1,\ F2,\ F3}_{\text{input}}$$

## Extensions for SIC/XE:

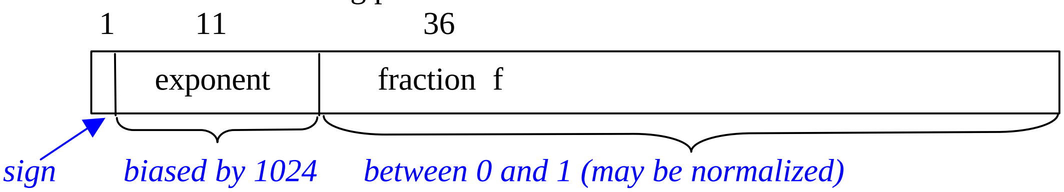
**Memory**: $2^{20}$ (1 M) bytes

**Added Registers**:

| mnemonic | number | |
|---|---|---|
| B | 3 | base register |
| S | 4 | general working register |
| T | 5 | general working register |
| F | 6 | floating point accumulator; it uses the 24 bits that could be R7 to provide a 48 bit register |

**Added Data formats**:

Numeric - 48 bit floating point



The actual exponent: exponent - 1024

value represented = (sgn) f × $2^{(\text{exponent - 1024})}$

**Instruction formats**: (4 in all)

1. 1-byte format (e.g., SIO, HIO, TIO, NORM) - not used in COP 3601

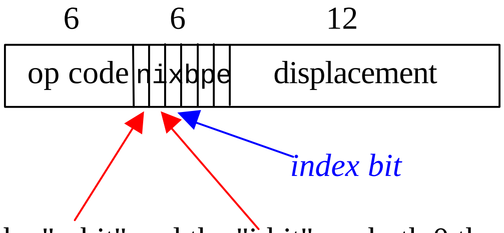
| 8 |
|---|
| op code |

2. 2-byte format

| 8 | 4 | 4 |
|---|---|---|
| op code | R1 | R2 |

The two addresses typically represent registers, so no memory access is needed to execute these.

3. 3-byte format



If the "n-bit" and the "i-bit" are both 0 then the instruction is interpreted as a simple SIC instruction.

4. 4-byte format

| 6 | 6 | | | | | | 20 |
|---|---|---|---|---|---|---|---|
| op code | n | i | x | b | p | e | address |

n = indirect bit
i = immediate bit
x = index bit
b = base bit
p = PC relative bit
e = extended bit

These bits alone or in combination determine variations of the instruction interpretation:

e = 0 ⇒ *3 byte format*
e = 1 ⇒ *4 byte format*

x = 1 ⇒ *indexed addressing*

b = 1 and p = 0 ⇒ *base/displacement* addressing
b = 0 and p = 1 ⇒ *PC relative* addressing

n = 1 and i = 1 ⇒ *direct* addressing
n = 1 and i = 0 ⇒ *indirect* addressing
n = 0 and i = 1 ⇒ *immediate* addressing

n = 0 and i = 0 ⇒ *simple SIC* interpretation
(so the last 15 bits is treated as an address,
including the *bpe* bits).

[The full collection of allowed interpretations is given in Appendix A of the course text.]